

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-151367

(43)Date of publication of application : 30.05.2000

(51)Int.Cl.

H03K 3/84  
G09C 1/00

(21)Application number : 10-325277

(71)Applicant : DENSO CORP

(22)Date of filing : 16.11.1998

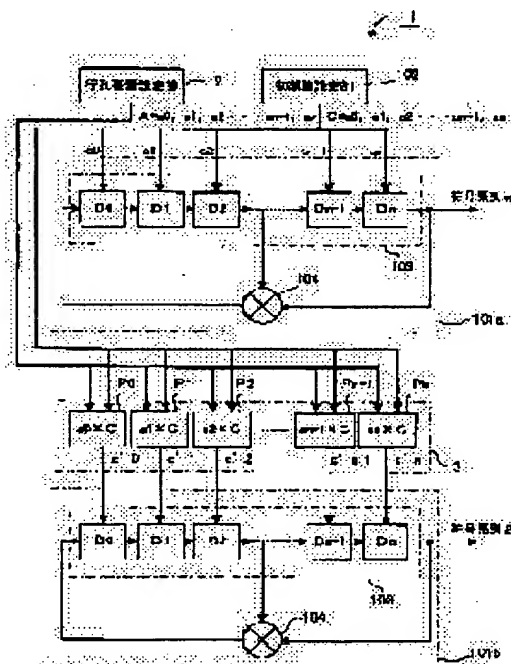
(72)Inventor : AONO TAKAYUKI  
ITO TAKAFUMI

## (54) CODE SERIES GENERATOR

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To shorten the time required up the start of the generation of a code series whose offset quantity is arbitrarily set and to decrease the circuit scale when the code series is generated.

**SOLUTION:** An initial value  $C$  is set in respective flip-flops  $D_0$  to  $D_n$  of the code series generator 101a and also set in computing elements  $P_0$  to  $P_n$  of a computing element group 3. Matrix elements  $a_0$  to  $a_n$  of an  $(n+1) \times (n+1)$  matrix  $A$  are set in the computing elements  $P_0$  to  $P_n$ . The computing elements  $P_0$  to  $P_n$  perform matrix operation between the initial value  $C$  and matrix elements  $a_0$  to  $a_n$  to obtain respective matrix operation results  $c'_0$  to  $c'_n$ . The matrix operation results  $c'_0$  to  $c'_n$  are set as initial values in respective flip-flops  $D_0$  to  $D_n$  of a code series generator 101b. When the same clock is supplied to the  $D_0$  to  $D_n$  of the code series generators 101a and 101b in this state, code series  $\alpha$  and  $\beta$  of  $M$  series having cycles  $2n+1-1$  are generated at the same time. Here, the matrix elements  $a_0$  to  $a_n$  are set, corresponding to the offset quantities of the code series  $\alpha$  and  $\beta$ .



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-151367

(P2000-151367A)

(43) 公開日 平成12年5月30日 (2000.5.30)

(51) Int.Cl.<sup>7</sup>

識別記号

F I

テーマコード(参考)

H 0 3 K 3/84

H 0 3 K 3/84

A 5 J 0 4 9

G 0 9 C 1/00

G 0 9 C 1/00

6 5 0 B 5 J 1 0 4

6 5 0

9 A 0 0 1

審査請求 未請求 請求項の数 6 O L (全 14 頁)

(21) 出願番号

特願平10-325277

(22) 出願日

平成10年11月16日 (1998. 11. 16)

(71) 出願人 000004260

株式会社デンソー

愛知県刈谷市昭和町1丁目1番地

(72) 発明者 青野 孝之

愛知県刈谷市昭和町1丁目1番地 株式会  
社デンソー内

(72) 発明者 伊藤 隆文

愛知県刈谷市昭和町1丁目1番地 株式会  
社デンソー内

(74) 代理人 100082500

弁理士 足立 勉

Fターム(参考) 5J049 AA00 AA18 AA33 CB01 CB06

5J104 AA18 FA04 NA04 NA08

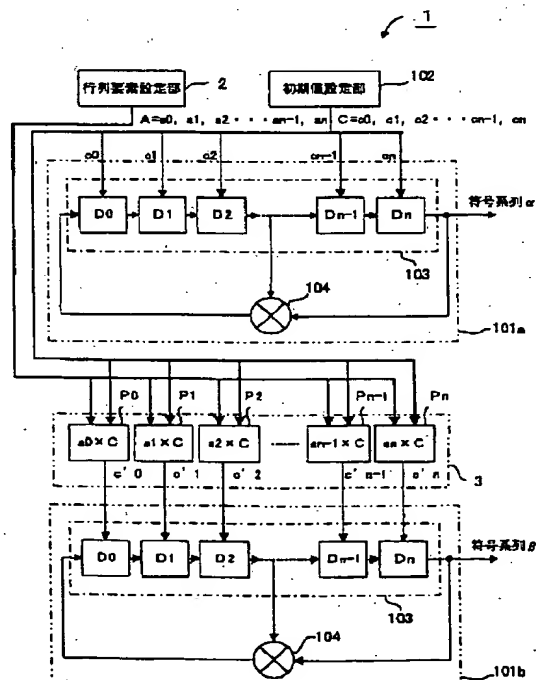
9A001 BB02 GG21

(54) 【発明の名称】 符号系列生成器

(57) 【要約】

【課題】 任意のオフセット量が設定された符号系列を生成する際に、その生成開始までに要する時間を短縮すると共に回路規模を小さくする。

【解決手段】 初期値Cは、符号系列生成器101aの各フリップフロップD0～Dnにセットされると共に、演算器群3の各演算器P0～Pnにセットされる。(n+1)×(n+1)行列Aの行列要素a0～anは各演算器P0～Pnにセットされる。各演算器P0～Pnは初期値Cと行列要素a0～anとの行列演算を行い、各行列演算結果c'0～c'nを算出する。各行列演算結果c'0～c'nは初期値として、符号系列生成器101bの各フリップフロップD0～Dnにセットされる。この状態で、各符号系列生成器101a、101bの各D0～Dnに同一のクロックを与えると、周期2<sup>n+1</sup>-1のM系列の各符号系列α、βが同時に生成される。尚、行列要素a0～anは各符号系列α、βのオフセット量に対応して設定されている。



## 【特許請求の範囲】

【請求項1】 第1シフトレジスタと、当該第1シフトレジスタの最終段の出力と任意の段の出力との排他的論理和をとり、当該排他的論理和を前記第1シフトレジスタの初段に帰還する第1排他的論理和回路とを備え、第1シフトレジスタの各段に初期値をセットし、第1シフトレジスタの各段にクロックを与えて、その各段の内容を順次次段にシフトさせることにより第1符号系列を生成し、当該第1符号系列をクロックに従って第1シフトレジスタの最終段から順次出力する第1符号系列生成器と、

第2シフトレジスタと、当該第2シフトレジスタの最終段の出力と任意の段の出力との排他的論理和をとり、当該排他的論理和を前記第2シフトレジスタの初段に帰還する第2排他的論理和回路とを備え、第2シフトレジスタの各段に初期値をセットし、第2シフトレジスタの各段にクロックを与えて、その各段の内容を順次次段にシフトさせることにより第2符号系列を生成し、当該第2符号系列をクロックに従って第2シフトレジスタの最終段から順次出力する第2符号系列生成器と、前記第2シフトレジスタの各段にセットされる初期値と、前記第1符号系列と前記第2符号系列とのオフセット量に対応して予め設定された行列要素から成る行列との行列演算を行い、その行列演算結果を前記第1シフトレジスタの各段の初期値としてセットする演算手段とを備えたことを特徴とする符号系列生成器。

【請求項2】 第1シフトレジスタと、当該第1シフトレジスタの最終段の出力と任意の段の出力との排他的論理和をとり、当該排他的論理和を前記第1シフトレジスタの初段に帰還する第1排他的論理和回路とを備え、第1シフトレジスタの各段に初期値をセットし、第1シフトレジスタの各段にクロックを与えて、その各段の内容を順次次段にシフトさせることにより第1符号系列を生成し、当該第1符号系列をクロックに従って第1シフトレジスタの最終段から順次出力する第1符号系列生成器と、

前記第1シフトレジスタの各段に対応する初期値と、前記第1符号系列の任意の位相に対応して予め設定された行列要素から成る行列との行列演算を行い、その行列演算結果を前記第1シフトレジスタの各段の初期値としてセットする演算手段とを備えたことを特徴とする符号系列生成器。

【請求項3】 請求項1または請求項2に記載の符号系列生成器において、前記演算手段は前記第1シフトレジスタの各段にそれぞれ対応する複数の演算器を備え、当該各演算器はそれぞれ前記第1シフトレジスタの各段に対応する前記行列の行列要素と前記初期値との行列演算を行うことを特徴とする符号系列生成器。

【請求項4】 請求項1または請求項2に記載の符号系

列生成器において、

前記演算手段は、

前記第1シフトレジスタの各段に対応する前記行列の行列要素と前記初期値との行列演算を各段毎に順次行う演算器と、

前記演算器が順次算出する前記第1シフトレジスタの各段毎の行列演算結果を、前記第1シフトレジスタの各段の初期値として当該各段毎に切り替えてセットする切替手段とを備えたことを特徴とする符号系列生成器。

【請求項5】 シフトレジスタと、当該シフトレジスタの最終段の出力と任意の段の出力との排他的論理和をとり、当該排他的論理和を前記シフトレジスタの初段に帰還する排他的論理和回路とを備え、シフトレジスタの各段に初期値をセットし、シフトレジスタの各段にクロックを与えて、その各段の内容を順次次段にシフトさせることにより第1符号系列を生成し、当該第1符号系列をクロックに従ってシフトレジスタの最終段から順次出力する符号系列生成器と、

前記シフトレジスタの各段の出力を初期値とし、当該初期値と、前記第1符号系列と第2符号系列とのオフセット量に対応して予め設定された行列要素から成る行列との行列演算を行うことにより、第2符号系列を生成する演算手段とを備えたことを特徴とする符号系列生成器。

【請求項6】 請求項1～5のいずれか1項に記載の符号系列生成器において、

前記行列を設定するための設定手段を備え、前記演算手段は当該設定手段により設定された任意の行列の対応する行列要素と前記初期値との行列演算を行うことを特徴とする符号系列生成器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は符号系列生成器に係り、詳しくは、最長系列(M系列)やゴールド(Gold)系列などの符号系列を生成する符号系列生成器に関するものである。

【0002】

【従来の技術】デジタル符号系列は、デジタル情報の秘匿や、スペクトラム拡散通信における信号帯域拡大のための拡散符号など、広い分野で利用されている。特に、M系列は、その周期が非常に長い上に、自己相関特性がインパルス的なため、ランダム符号や雑音源などとして利用されている。

【0003】図4は、M系列を生成するために従来より一般に使用されている符号系列生成器101の概略構成を表すブロック図である。M系列の符号系列生成器101は、初期値設定部102、シフトレジスタ103、排他的論理和演算回路104から構成されている。

【0004】 $n+1$ ビット長のシフトレジスタ103は、各フリップフロップD0～Dnが $n+1$ 段シリーズに接続されて構成されている。排他的論理和演算回路1

04は、シフトレジスタ103における最後段( $n+1$ 段目)のフリップフロップD $n$ の出力と、生成すべき符号系列に対応した任意の段のフリップフロップ(例えば、3段目フリップフロップD2)の出力との排他的論理和をとり、その結果を初段のフリップフロップD0に帰還する。

【0005】ユーザが初期値設定部102を用いて設定した初期値C( $=c_0, c_1, c_2 \dots, c_{n-1}, c_n$ )は、それぞれ各フリップフロップD0~D $n$ にセットされる。そして、各フリップフロップD0~D $n$ にクロック(図示略)を与えて、その内容を順次次段のフリップフロップにシフトさせることにより、周期 $2^{n+1}-1$ のM系列の符号系列が生成され、その符号系列はクロックに従ってフリップフロップD $n$ から順次出力される。

【0006】このように構成された符号系列生成器101において、任意の初期値Cをセットした後に任意の個数のクロックを与えると、そのクロック数分だけシフトされた符号系列が生成される。つまり、与えるクロック数を変更することにより、そのクロック数に対応した任意の位相から符号系列を生成することができる。これを利用して、符号系列生成器101を2つ用いることにより、任意のクロック数分だけ位相のずれた2つの符号系列を同時に生成する方法が提案されている。

【0007】図5は、任意のクロック数分だけ位相のずれた2つのM系列符号を生成することが可能な符号系列生成器201の概略構成を表すブロック図である。符号系列生成器201は、初期値設定部102を共有する2つの符号系列生成器101(以下、2つを区別するため「101a」「101b」と符号を付す)から構成されている。

【0008】図6は、符号系列生成器201の動作を表すタイミングチャートである。符号系列生成器201を動作させるには、まず、各符号系列生成器101a、101bを構成する各フリップフロップD0~D $n$ に対して、ユーザが初期値設定部102を用いて設定した同じ初期値C( $=c_0, c_1, c_2 \dots, c_{n-1}, c_n$ )をセットする。次に、符号系列生成器101bの各フリップフロップD0~D $n$ に任意の個数のクロックを予め与えておき、その内容を順次次段のフリップフロップにシフトさせておく。その後、各符号系列生成器101a、101bの各フリップフロップD0~D $n$ に同一のクロックを与える。その結果、符号系列生成器101aの生成する符号系列 $\alpha$ に対して、符号系列生成器101bからは予め与えておいたクロック数分だけ位相が進んだ符号系列 $\beta$ が生成される。つまり、各符号系列生成器101a、101bの生成する各符号系列 $\alpha, \beta$ には、符号系列生成器101bに予め与えておいたクロック数分のオフセット量が生じることになる。

【0009】ここで、各符号系列生成器101a、101bの各符号系列 $\alpha, \beta$ のオフセット量を大きくするには、符号系列生成器101bに予め与えておくクロック数を多くしなければならず、初期値Cをセットしてから各符号系列生成器101a、101bから位相のずれた符号系列 $\alpha, \beta$ が生成開始されるまでの間に、非常に長い時間を要する。

【0010】ところで、符号系列生成器101から任意の位相の符号系列を生成する場合においても、周期が非常に長い符号系列における任意の位相から符号系列を生成するには、予め与えておくクロック数を多くしなければならず、初期値Cをセットしてから所望の符号系列が生成開始されるまでの間に非常に長い時間を要する。

【0011】

【発明が解決しようとする課題】そこで、特開平8-18550号公報に開示されるように、レジスタに格納された $n$ ビット長のデータXに $(n+1) \times (n+1)$ 行列Aで規定された線形変換を繰り返して実行することにより、符号系列 $Y = A^m X$ ( $m$ は正の整数)を生成する符号系列生成器が提案されている。この符号系列生成器は、予め定めた $t+1$ 個の正の整数 $p_j$ ( $j=0, 1, 2, \dots, t$ )のそれぞれについて行列Aの $p_j$ 乗( $A^{p_j}$ )を記憶する記憶手段と、前記整数 $m$ を前記整数 $p_j$ の一次式 $m = k_0 + k_1 p_1 + k_2 p_2 + \dots + k_t p_t$ ( $k_0, k_1, k_2, \dots, k_t$ は0または整数)の形に分解する手段と、前記 $k_0, k_1, k_2, \dots, k_t$ の内ではゼロでないものについて前記記憶手段から対応する行列を順次選択する選択手段と、前記 $k_0, k_1, k_2, \dots, k_t$ の内ではゼロでないものについて行列演算 $Y = (A^{p_0})^{k_0} (A^{p_1})^{k_1} (A^{p_2})^{k_2} \dots (A^{p_t})^{k_t} X$ を実行する演算手段とを備えている。そのため、選択手段により選択された複数の行列を組み合わせる使用することにより、高速にレジスタ状態を設定し、任意の位相から符号系列を生成することができる。

【0012】しかし、同公報に記載の符号系列生成器では、前記選択手段により前記記憶手段から対応する行列を組み合わせる順次選択させるための制御が複雑であり、その制御を行うために大規模な回路構成の制御回路を要するという問題があった。本発明は上記問題点を解決するためになされたものであって、その目的は、任意のオフセット量が設定された同一系列の複数の符号系列を生成する際に、符号系列の生成開始までに要する時間を短縮することが可能で回路規模の小さな符号系列生成器を提供することにある。また、本発明の別の目的は、任意の位相から符号系列を生成する際に、符号系列の生成開始までに要する時間を短縮することが可能で回路規模の小さな符号系列生成器を提供することにある。

【0013】

【課題を解決するための手段】かかる目的を達成するためになされた請求項1に記載の発明は、第1符号系列生成器、第2符号系列生成器、演算手段を備えている。第

1 符号系列生成器は、第1 シフトレジスタと、当該第1 シフトレジスタの最終段の出力と任意の段の出力との排他的論理和をとり、当該排他的論理和を前記第1 シフトレジスタの初段に帰還する第1 排他的論理和回路とを備え、第1 シフトレジスタの各段に初期値をセットし、第1 シフトレジスタの各段にクロックを与えて、その各段の内容を順次次段にシフトさせることにより第1 符号系列を生成し、当該第1 符号系列をクロックに従って第1 シフトレジスタの最終段から順次出力する。第2 符号系列生成器は、第2 シフトレジスタと、当該第2 シフトレジスタの最終段の出力と任意の段の出力との排他的論理和をとり、当該排他的論理和を前記第2 シフトレジスタの初段に帰還する第2 排他的論理和回路とを備え、第2 シフトレジスタの各段に初期値をセットし、第2 シフトレジスタの各段にクロックを与えて、その各段の内容を順次次段にシフトさせることにより第2 符号系列を生成し、当該第2 符号系列をクロックに従って第2 シフトレジスタの最終段から順次出力する。演算手段は、前記第2 シフトレジスタの各段にセットされる初期値と、前記第1 符号系列と前記第2 符号系列とのオフセット量に対応して予め設定された行列要素から成る行列との行列演算を行い、その行列演算結果を前記第1 シフトレジスタの各段の初期値としてセットする。

【0014】従って、請求項1に記載の発明によれば、第1 符号系列と第2 符号系列とのオフセット量を設定するために第1 符号系列生成器にのみ予めクロックを与えておく必要がない。また、演算手段における行列演算に要する時間は、従来の符号系列生成器において第1、第2 符号系列のオフセット量を設定するために第1 符号系列生成器に予め与えておくクロック数分に相当する時間に比べればごく短い。そのため、請求項1に記載の発明によれば、初期値をセットしてから任意のオフセット量が設定された第1、第2 符号系列の生成開始までに要する時間を短縮することができる。

【0015】次に、請求項2に記載の発明は、第1 符号生成器および演算手段を備えている。第1 符号生成器は、第1 シフトレジスタと、当該第1 シフトレジスタの最終段の出力と任意の段の出力との排他的論理和をとり、当該排他的論理和を前記第1 シフトレジスタの初段に帰還する第1 排他的論理和回路とを備え、第1 シフトレジスタの各段に初期値をセットし、第1 シフトレジスタの各段にクロックを与えて、その各段の内容を順次次段にシフトさせることにより第1 符号系列を生成し、当該第1 符号系列をクロックに従って第1 シフトレジスタの最終段から順次出力する。演算手段は、前記第1 シフトレジスタの各段に対応する初期値と、前記第1 符号系列の任意の位相に対応して予め設定された行列要素から成る行列との行列演算を行い、その行列演算結果を前記第1 シフトレジスタの各段の初期値としてセットする。

【0016】従って、請求項2に記載の発明によれば、

任意の位相の符号系列を生成する際に、第1 符号系列生成器に予めクロックを与えておく必要がない。また、演算手段における行列演算に要する時間は、従来の符号系列生成器において任意の位相の符号系列を生成する際に第1 符号系列生成器に予め与えておくクロック数分に相当する時間に比べればごく短い。そのため、請求項2に記載の発明によれば、初期値をセットしてから任意の位相の符号系列の生成開始までに要する時間を短縮することができる。

【0017】ところで、請求項3に記載の発明のように、請求項1 または請求項2 に記載の符号系列生成器において、前記演算手段は前記第1 シフトレジスタの各段にそれぞれ対応する複数の演算器を備え、当該各演算器はそれぞれ前記第1 シフトレジスタの各段に対応する前記行列の行列要素と前記初期値との行列演算を行うようにしてもよい。

【0018】このようにすれば、第1 シフトレジスタの各段に対応する前記行列の行列要素と前記初期値との行列演算がそれぞれ別個の演算器にて同時に算出されるため、初期値をセットしてから任意の位相の符号系列の生成開始までに要する時間をさらに短縮することができる。

【0019】また、請求項4に記載の発明のように、請求項1 または請求項2 に記載の符号系列生成器において、前記演算手段は、前記第1 シフトレジスタの各段に対応する前記行列の行列要素と前記初期値との行列演算を各段毎に順次行う演算器と、前記演算器が順次算出する前記第1 シフトレジスタの各段毎の行列演算結果を、前記第1 シフトレジスタの各段の初期値として当該各段毎に切り替えてセットする切替手段とを備えるようにしてもよい。

【0020】このようにすれば、1つの演算器を設けるだけで、第1 シフトレジスタの各段に対応する前記行列の行列要素と前記初期値との行列演算を行い、その行列演算結果を第1 シフトレジスタの各段の初期値として当該各段毎に切り替えてセットすることができる。従って、第1 シフトレジスタの各段にそれぞれ対応する複数の演算器を設ける場合に比べて、演算器の数を減らせる分だけ全体の回路規模を小さくすることができる。

【0021】次に、請求項5に記載の発明は、符号生成器および演算手段を備えている。符号生成器は、シフトレジスタと、当該シフトレジスタの最終段の出力と任意の段の出力との排他的論理和をとり、当該排他的論理和を前記シフトレジスタの初段に帰還する排他的論理和回路とを備え、シフトレジスタの各段に初期値をセットし、シフトレジスタの各段にクロックを与えて、その各段の内容を順次次段にシフトさせることにより第1 符号系列を生成し、当該第1 符号系列をクロックに従ってシフトレジスタの最終段から順次出力する。演算手段は、前記シフトレジスタの各段の出力を初期値とし、当該初

期値と、前記第1符号系列と第2符号系列とのオフセット量に対応して予め設定された行列要素から成る行列との行列演算を行うことにより、第2符号系列を生成する。

【0022】従って、請求項5に記載の発明によれば、第1符号系列と第2符号系列とのオフセット量を設定するために符号系列生成器に予めクロックを与えておく必要がない。また、演算手段における行列演算に要する時間は、従来の符号系列生成器において第1、第2符号系列のオフセット量を設定するために符号系列生成器に予め与えておくクロック数分に相当する時間に比べればごく短い。そのため、請求項1に記載の発明によれば、初期値をセットしてから任意のオフセット量が設定された第1、第2符号系列の生成開始までに要する時間を短縮することができる。また、演算手段においては、演算器を1つ設けるだけで第2符号系列を生成することができる。そして、シフトレジスタおよび排他的論理和回路を備えた符号系列生成器を1つ設けるだけで、第1、第2符号系列を生成することができる。そのため、請求項1に記載の発明のように第1、第2符号系列生成器を設ける場合に比べて、全体の回路規模を小さくすることができる。

【0023】ところで、請求項6に記載の発明のように、請求項1～5のいずれか1項に記載の符号系列生成器において、前記行列を設定するための設定手段を備え、前記演算手段は当該設定手段により設定された任意の行列の対応する行列要素と前記初期値との行列演算を行うようにしてもよい。

【0024】このようにすれば、第1、第2符号系列のオフセット量を変更する際や符号系列の位相を変更する際に、設定手段を用いて設定する行列の各行列要素を適宜変更して容易に対応することができる。つまり、第1、第2符号系列のオフセット量に対応して離散的に離れた第1シフトレジスタのレジスタ状態を表す行列を予め設定するのが簡単になるため、オフセット量の大きな第1、第2符号系列を生成する場合や、周期が非常に長い符号系列における任意の位相から符号系列を生成する場合に特に有効である。

【0025】尚、以下に述べる発明の実施の形態において、特許請求の範囲または課題を解決するための手段に記載の「第1符号系列生成器」は符号系列生成器101bに相当し、同じく「第2符号系列生成器」は符号系列生成器101aに相当し、同じく「第1符号系列」は符号系列 $\beta$ に相当し、同じく「第2符号系列」は符号系列 $\alpha$ に相当し、「演算手段」は演算器群3または演算器14および切替回路15から構成され、同じく「設定手段」は行列要素設定部2またはメモリ12および読出回路13から構成される。

【0026】

【発明の実施の形態】（第1実施形態）以下、本発明を

具体化した第1実施形態を図面と共に説明する。尚、本第1実施形態において、図5に示した従来の形態と同じ構成部材については符号を等しくしてその詳細な説明を省略する。

【0027】図1は、第1実施形態の符号系列生成器1の概略構成を表すブロック図である。符号系列生成器1は、初期値設定部102、符号系列生成器101a、101b、行列要素設定部2、演算器群3から構成されている。

【0028】演算器群3は、 $n+1$ 個の演算器 $P_0 \sim P_n$ から構成されている。ユーザが初期値設定部102を用いて設定した初期値 $C (=c_0, c_1, c_2, \dots, c_{n-1}, c_n)$ は、符号系列生成器101aのシフトレジスタ103を構成する各フリップフロップ $D_0 \sim D_n$ にセットされると共に、演算器群3の各演算器 $P_0 \sim P_n$ にセットされる。すなわち、初期値 $c_0$ はフリップフロップ $D_0$ にセットされ、初期値 $c_1$ はフリップフロップ $D_1$ にセットされ……、初期値 $c_n$ はフリップフロップ $D_n$ にセットされるといった具合に、 $n+1$ 段シリーズに接続された各フリップフロップ $D_0 \sim D_n$ に対し、そのフリップフロップの段数の昇順（符号の昇順）に対応した各初期値 $c_0 \sim c_n$ が昇順に1つずつセットされる。また、各演算器 $P_0 \sim P_n$ には初期値 $C$ （全ての初期値 $c_0 \sim c_n$ ）がセットされる。

【0029】ユーザが行列要素設定部2を用いて設定した $(n+1) \times (n+1)$ 行列 $A$ の行列要素 $a_0, a_1, a_2, \dots, a_{n-1}, a_n$ は、演算器群3の各演算器 $P_0 \sim P_n$ にセットされる。すなわち、行列要素 $a_0$ は演算器 $P_0$ にセットされ、行列要素 $a_1$ は演算器 $P_1$ にセットされ……、行列要素 $a_n$ は演算器 $P_n$ にセットされるといった具合に、各演算器 $P_0 \sim P_n$ に対し、その演算器の符号の昇順に対応した各行列要素 $a_0 \sim a_n$ が昇順に1つずつセットされる。

【0030】各演算器 $P_0 \sim P_n$ はそれぞれ、セットされた初期値 $C (=c_0 \sim c_n)$ と行列要素 $a_0 \sim a_n$ との行列演算を行い、各行列演算結果 $c'_0 \sim c'_n$ を算出する。すなわち、演算器 $P_0$ は初期値 $C$ と行列要素 $a_0$ との行列演算結果 $c'_0$ を算出し、演算器 $P_1$ は初期値 $C$ と行列要素 $a_1$ との行列演算結果 $c'_1$ を算出し……、演算器 $P_n$ は初期値 $C$ と行列要素 $a_n$ との行列演算結果 $c'_n$ を算出するといった具合に、各演算器 $P_0 \sim P_n$ はそれぞれ各行列演算結果 $c'_0 \sim c'_n$ を算出する。

【0031】その各行列演算結果 $c'_0 \sim c'_n$ が初期値として、符号系列生成器101bのシフトレジスタ103を構成する各フリップフロップ $D_0 \sim D_n$ にセットされる。すなわち、行列演算結果 $c'_0$ はフリップフロップ $D_0$ にセットされ、行列演算結果 $c'_1$ はフリップフロップ $D_1$ にセットされ……、行列演算結果 $c'_n$ はフリップフロップ $D_n$ にセットされるといった具合

に、 $n+1$  段シリーズに接続された各フリップフロップ  $D0 \sim Dn$  に対し、そのフリップフロップの段数の昇順（符号の昇順）に対応した各行列演算結果（初期値） $c'0 \sim c'n$  が昇順に1つつセットされる。

【0032】このように、符号系列生成器101aの各フリップフロップ  $D0 \sim Dn$  にそれぞれ各初期値  $c'0 \sim c'n$  がセットされると共に、符号系列生成器101bの各フリップフロップ  $D0 \sim Dn$  にそれぞれ各初期値  $c'0 \sim c'n$  がセットされた状態で、各符号系列生成器101a、101bの各フリップフロップ  $D0 \sim Dn$  に同一のクロックを与える。その結果、符号系列生成器101aから周期  $2^{n+1}-1$  のM系列の符号系列 $\alpha$ が生成されるのと同時に、符号系列生成器101bから周期  $2^{n+1}-1$  のM系列の符号系列 $\beta$ が生成される。

【0033】ここで、各符号系列 $\alpha$ 、 $\beta$ に所望のオフセット量を生じさせるには、各初期値  $c'0 \sim c'n$  と各初期値  $c'0 \sim c'n$  に当該オフセット量を設定しておけばよい。それには、各初期値  $c'0$ 、 $c'0$  に当該オフセット量が設定されるように行列要素  $a_0$  を設定し、各初期値  $c'1$ 、 $c'1$  に当該オフセット量が設定されるように行列要素  $a_1$  を設定し……、各初期値  $c'n$ 、 $c'n$  に当該オフセット量が設定されるように行列要素  $a_n$  を設定するといった具合に、当該オフセット量に対応した各行列要素  $a_0 \sim a_n$  から成る行列Aを設定しておけばよい。

【0034】つまり、図5に示す従来の符号系列生成器201においては、符号系列生成器101bの各フリップフロップ  $D0 \sim Dn$  にのみ予め任意の個数のクロックを与えておき、その後で各符号系列生成器101a、101bの各フリップフロップ  $D0 \sim Dn$  に同一のクロックを与えることにより、符号系列生成器101bに予め与えておいたクロック数分のオフセット量が各符号系列 $\alpha$ 、 $\beta$ に生じるようにしている。そのため、図6に示すように、初期値Cをセットしてから各符号系列生成器101a、101bからオフセット量を有した符号系列 $\alpha$ 、 $\beta$ が生成開始されるまでの間に、符号系列生成器101bに予め与えておくクロック数分に相当する時間を要することになる。

【0035】それに対して、本第1実施形態の符号系列生成器1においては、各符号系列 $\alpha$ 、 $\beta$ のオフセット量に対応した各行列要素  $a_0 \sim a_n$  から成る行列Aを行列要素設定部2を用いて予め設定しておき、符号系列生成器101aに与える初期値Cと各行列要素  $a_0 \sim a_n$  との行列演算結果を各演算器  $P0 \sim Pn$  にて算出し、その行列演算結果を符号系列生成器101bの初期値  $c'0 \sim c'n$  として与えている。

【0036】そのため、符号系列生成器1では、各符号系列 $\alpha$ 、 $\beta$ のオフセット量を設定するために符号系列生成器101bにのみ予めクロックを与えておく必要がない。また、各演算器  $P0 \sim Pn$  における行列演算に要す

る時間は、従来の符号系列生成器201において各符号系列 $\alpha$ 、 $\beta$ のオフセット量を設定するために符号系列生成器101bに予め与えておくクロック数分に相当する時間に比べればごく短い。

【0037】従って、符号系列生成器1によれば、初期値Cをセットしてから任意のオフセット量が設定された各符号系列 $\alpha$ 、 $\beta$ の生成開始までに要する時間を短縮することができる。そして、各符号系列 $\alpha$ 、 $\beta$ のオフセット量を変更するには、行列要素設定部2を用いて設定する行列Aの各行列要素  $a_0 \sim a_n$  を適宜変更すればよい。つまり、各符号系列 $\alpha$ 、 $\beta$ のオフセット量に対応して離散的に離れたシフトレジスタ103のレジスタ状態（各フリップフロップ  $D0 \sim Dn$  の内容）を表す行列Aを予め設定しておけばよい。従って、符号系列生成器1は、オフセット量の大きな各符号系列 $\alpha$ 、 $\beta$ を生成する場合に特に有効である。

【0038】また、符号系列生成器1の制御動作は、従来の符号系列生成器201の制御動作とほぼ同じであり、その制御を行うための制御回路（図示略）の回路規模が増大することはない。

（第2実施形態）次に、本発明を具体化した第2実施形態を図面と共に説明する。尚、本第2実施形態において、第1実施形態と同じ構成部品については符号を等しくしてその詳細な説明を省略する。

【0039】図2は、第2実施形態の符号系列生成器1の概略構成を表すブロック図である。符号系列生成器11は、初期値設定部102、符号系列生成器101a、101b、メモリ12、読出回路13、演算器14、切替回路15から構成されている。

【0040】ユーザが初期値設定部102を用いて設定した初期値C（ $=c'0 \sim c'n$ ）は、符号系列生成器101aのシフトレジスタ103を構成する各フリップフロップ  $D0 \sim Dn$  にセットされると共に、演算器14にセットされる。メモリ12には、各行列要素  $a_0 \sim a_n$  から成る  $(n+1) \times (n+1)$  行列Aが記憶されている。

【0041】読出回路13は、メモリ12に記憶されている各行列要素  $a_0 \sim a_n$  を符号の昇順に順次読み出し、その読み出した行列要素  $a_m$ （ $m=0, 1, 2, \dots, n$ ）を演算器14にセットする。演算器14は、セットされた行列要素  $a_m$  と初期値C（ $=c'0 \sim c'n$ ）との行列演算を行い、各行列演算結果  $c'0 \sim c'n$  を順次算出する。すなわち、演算器14は、まず、初期値Cと行列要素  $a_0$  との行列演算結果  $c'0$  を算出し、次に、初期値Cと行列要素  $a_1$  との行列演算結果  $c'1$  を算出し……、最後に、初期値Cと行列要素  $a_n$  との行列演算結果  $c'n$  を算出するといった具合に、各行列演算結果  $c'0 \sim c'n$  を順次算出する。

【0042】切替回路15は、演算器14の算出した各行列演算結果  $c'0 \sim c'n$  を1つつ選択切替し、そ

の選択切替した行列演算結果を初期値として、符号系列生成器101bのシフトレジスタ103を構成する各フリップフロップD0～Dnにセットする。すなわち、行列演算結果 $c'0$ はフリップフロップD0にセットされ、行列演算結果 $c'1$ はフリップフロップD1にセットされ……、行列演算結果 $c'n$ はフリップフロップDnにセットされるといった具合に、 $n+1$ 段シリーズに接続された各フリップフロップD0～Dnに対し、そのフリップフロップの段数の昇順(符号の昇順)に対応した各行列演算結果(初期値) $c'0 \sim c'n$ が昇順に1つつ順次セットされる。

【0043】このように、符号系列生成器101aの各フリップフロップD0～Dnにそれぞれ各初期値 $c0 \sim cn$ がセットされると共に、符号系列生成器101bの各フリップフロップD0～Dnにそれぞれ各初期値 $c'0 \sim c'n$ がセットされた状態で、各符号系列生成器101a、101bの各フリップフロップD0～Dnに同一のクロックを与える。その結果、符号系列生成器101aから周期 $2^{n+1}-1$ のM系列の符号系列 $\alpha$ が生成されるのと同時に、符号系列生成器101bから周期 $2^{n+1}-1$ のM系列の符号系列 $\beta$ が生成される。

【0044】従って、本第2実施形態の符号系列生成器11によれば、第1実施形態の符号系列生成器1と同様の作用・効果を得ることができる。但し、符号系列生成器11においては、メモリ12に記憶された行列Aの各行列要素 $a0 \sim an$ を読出回路13により順次読み出して演算器14にセットし、演算器14の算出した各行列演算結果(初期値) $c'0 \sim c'n$ を切替回路15により選択切替して符号系列生成器101bの各フリップフロップD0～Dnにセットするようにしている。つまり、読出回路13を設けることにより、1つの演算器14のみで $n+1$ ビットの初期値 $c'0 \sim c'n$ を生成することが可能になる。そして、切替回路15を設けることにより、1つの演算器14の算出した各初期値 $c'0 \sim c'n$ を符号系列生成器101bの各フリップフロップD0～Dnにそれぞれセットすることが可能になる。

【0045】それに対して、第1実施形態の符号系列生成器1においては、各初期値 $c'0 \sim c'n$ をそれぞれ別個に算出し、その各初期値 $c'0 \sim c'n$ を符号系列生成器101bの各フリップフロップD0～Dnにそれぞれセットするため、 $n+1$ 個の演算器P0～Pnを設けている。

【0046】従って、符号系列生成器11によれば、符号系列生成器1に比べて、演算器の数を減らせる分だけ全体の回路規模を小さくすることができる。但し、符号系列生成器11では、まず、メモリ12に記憶された各行列要素 $a0 \sim an$ を読出回路13により順次読み出し、次に、その行列要素に基づいて演算器14により初期値 $c'0 \sim c'n$ を順次算出し、続いて、その初期値を切替回路15により符号系列生成器101bの各フリ

ップフロップD0～Dnに順次セットしている。そのため、これらの動作(読出回路13、演算器14、切替回路15の動作)に要する時間分だけ、符号系列生成器1に比べて、任意のオフセット量が設定された各符号系列 $\alpha$ 、 $\beta$ の生成開始までに要する時間が長くなる。

【0047】しかし、前記動作(読出回路13、演算器14、切替回路15の動作)に要する時間は、従来の符号系列生成器201において各符号系列 $\alpha$ 、 $\beta$ のオフセット量を設定するために符号系列生成器101bに予め与えておくクロック数分に相当する時間に比べればごく短い。従って、符号系列生成器11においても、従来の符号系列生成器201に比べれば、任意のオフセット量が設定された各符号系列 $\alpha$ 、 $\beta$ の生成開始までに要する時間を短縮することができる。

【0048】また、読出回路13および切替回路15の動作は予め定められた手順を順次行うだけであり、その動作の制御は単純であるため、その制御を行うための制御回路(図示略)の回路規模は、前記公報(特開平8-18550号)に記載の符号系列生成器の制御回路の回路規模に比べて小さくなる。従って、符号系列生成器11によれば、前記公報に記載の符号系列生成器に比べて、前記制御回路の回路規模が小さくなる分だけ全体の回路規模を縮小することができる。

【0049】(第3実施形態)次に、本発明を具体化した第3実施形態を図面と共に説明する。尚、本第2実施形態において、第2実施形態と同じ構成部材については符号を等しくしてその詳細な説明を省略する。

【0050】図3は、第3実施形態の符号系列生成器21の概略構成を表すブロック図である。符号系列生成器21は、初期値設定部102、符号系列生成器101、メモリ12、読出回路13、演算器14、切替回路15から構成されている。

【0051】ユーザが初期値設定部102を用いて設定した初期値 $C (=c0 \sim cn)$ は、符号系列生成器101のシフトレジスタ103を構成する各フリップフロップD0～Dnにセットされる。そして、各フリップフロップD0～Dnにクロック(図示略)を与えて、その内容を順次次段のフリップフロップにシフトさせることにより、周期 $2^{n+1}-1$ のM系列の符号系列 $\alpha$ が生成され、その符号系列はクロックに従ってフリップフロップDnから順次出力される。このとき、各フリップフロップD0～Dnの出力は初期値 $C$ として演算器14にセットされる。

【0052】メモリ12には、各行列要素 $a0 \sim an$ から成る $(n+1) \times (n+1)$ 行列Aが記憶されている。読出回路13は、メモリ12に記憶されている各行列要素 $a0 \sim an$ を符号の昇順に順次読み出し、その読み出した行列要素 $a_m (m=0, 1, 2, \dots, n)$ を演算器14にセットする。

【0053】演算器14は、セットされた行列要素 $a_m$



と初期値C（各フリップフロップD0～Dnの出力）との行列演算を行い、各行列演算結果 $c'0 \sim c'n$ を順次算出することにより、各行列演算結果 $c'0 \sim c'n$ から成る符号系列 $\beta$ を生成する。

【0054】このように、本第3実施形態の符号系列生成器21において、メモリ12、読出回路13、演算器14の動作については、第2実施形態の符号系列生成器11と同じである。つまり、符号系列生成器11においては、メモリ12、読出回路13、演算器14、切替回路15を用いて各初期値 $c'0 \sim c'n$ を生成した後、その各初期値 $c'0 \sim c'n$ を符号系列生成器101bの各フリップフロップD0～Dnにセットすることにより、符号系列生成器101bから符号系列 $\beta$ を生成している。

【0055】それに対して、符号系列生成器21においては、符号系列生成器101の各フリップフロップD0～Dnの出力を初期値Cとして演算器14にセットし、演算器14の算出した各行列演算結果 $c'0 \sim c'n$ を符号系列 $\beta$ としている。従って、符号系列生成器21において、符号系列生成器11と異なるのは、初期値Cの生成方法だけであるため、その他の点については符号系列生成器11と同様の作用・効果を得ることができる。そして、符号系列生成器21によれば、2つの符号系列生成器101a、101bを設ける符号系列生成器11に比べて、1つの符号系列生成器101を設けるだけでよいと、全体の回路規模を小さくすることができる。

【0056】尚、本発明は上記各実施形態に限定されるものではなく、以下のように変更してもよく、その場合でも、上記各実施形態と同様の作用および効果を得ることができる。

(1) 第1実施形態において、符号系列生成器1から符号系列生成器101aを省き、符号系列生成器101bから任意の位相の符号系列 $\beta$ を生成するように変更してもよい。この場合は、行列要素設定部2を用いて、任意の位相に対応した各行列要素 $a0 \sim an$ から成る行列Aを予め設定しておく。このようにすれば、周期が非常に長い符号系列における任意の位相から符号系列を生成する場合でも、初期値Cをセットしてから所望の符号系列 $\beta$ が生成開始されるまでの時間を短縮することが可能であり、周期が長い符号系列を生成する場合に特に有効である。

【0057】(2) 第2実施形態において、符号系列生成器11から符号系列生成器101aを省き、符号系列生成器101bから任意の位相の符号系列 $\beta$ を生成するように変更してもよい。この場合は、任意の位相に対応した各行列要素 $a0 \sim an$ から成る行列Aをメモリ12に記憶させておく。このようにすれば、周期が非常に長い符号系列における任意の位相から符号系列を生成する場合でも、初期値Cをセットしてから所望の符号系列 $\beta$ が生成開始されるまでの時間を短縮することが可能であ

り、周期が長い符号系列を生成する場合に特に有効である。

【0058】(3) 第2、第3実施形態において、メモリ12に1種類の行列Aを記憶させておくだけでなく、各符号系列 $\alpha$ 、 $\beta$ のオフセット量に対応して各行列要素が適宜設定された複数の行列を記憶させておき、読出回路13により任意の行列の行列要素を読み出すようにしてもよい。すなわち、オフセット量 $x$ に対応した行列Xと、オフセット量 $y$ に対応した行列Yとをメモリ12に記憶させておき、各符号系列 $\alpha$ 、 $\beta$ にオフセット量 $x$ が必要な場合は読み出し回路13により行列Xの各行列要素を読み出し、各符号系列 $\alpha$ 、 $\beta$ にオフセット量 $y$ が必要な場合は読み出し回路13により行列Yの各行列要素を読み出すようにする。このようにすれば、オフセット量が異なる各符号系列 $\alpha$ 、 $\beta$ を任意に生成することができる。

【0059】(4) 第1実施形態において、行列要素設定部2、演算器群3、符号系列生成器101bを複数組設け、それら各組を異なる行列要素から成る行列を用いて並列動作させることにより、相互にオフセット量が設定された3つ以上の符号系列を同時に生成可能にしてもよい。

【0060】(5) 第2実施形態において、読出回路13、演算器14、切替回路15、符号系列生成器101bを複数組設け、それら各組を異なる行列要素から成る行列を用いて並列動作させることにより、相互にオフセット量が設定された3つ以上の符号系列を同時に生成可能にしてもよい。

【0061】(6) 第3実施形態において、読出回路13および演算器14を複数組設け、それら各組を異なる行列要素から成る行列を用いて並列動作させることにより、相互にオフセット量が設定された3つ以上の符号系列を同時に生成可能にしてもよい。

【0062】(7) 本発明はM系列ではなくゴールド系列の符号系列生成器に適用してもよい。図7は、ゴールド系列を生成するために従来より一般に使用されている符号系列生成器301の概略構成を表すブロック図である。

【0063】ゴールド系列の符号系列生成器301は、初期値設定部102、符号系列生成器101c、101d、排他的論理和302から構成されている。符号系列生成器101cの排他的論理和回路104は、シフトレジスタ103における最終段のフリップフロップDnの出力と、生成すべき符号系列に対応した $x$ 段目( $x=0, 1, 2, \dots, n$ )のフリップフロップD $x$ の出力との排他的論理和をとり、その結果を初段のフリップフロップD0に帰還する。また、符号系列生成器101dの排他的論理和回路104は、シフトレジスタ103における最終段のフリップフロップDnの出力と、生成すべき符号系列に対応した $y$ 段目( $y=0, 1, 2, \dots, n, y$

$\neq x$ ) のフリップフロップ  $Dy$  の出力との排他的論理和をとり、その結果を初段のフリップフロップ  $D0$  に帰還する。

【0064】排他的論理和302は、符号系列生成器101cから生成された符号系列 $\alpha$ と、符号系列生成器101dから生成された符号系列 $\beta$ との排他的論理和をとり、ゴールド系列の符号系列を生成する。第1実施形態において、M系列ではなくゴールド系列の符号系列を生成するには、図1に示す各符号系列生成器101a, 101bをそれぞれ符号系列生成器301に置き換えればよい。

【0065】また、第2実施形態において、M系列ではなくゴールド系列の符号系列を生成するには、図2に示す各符号系列生成器101a, 101bをそれぞれ符号系列生成器301に置き換えればよい。また、第3実施形態において、M系列ではなくゴールド系列の符号系列を生成するには、図3に示す符号系列生成器101を符号系列生成器301に置き換えればよい。

【図面の簡単な説明】

【図1】本発明を具体化した第1実施形態の概略構成を表すブロック図。

【図2】本発明を具体化した第2実施形態の概略構成を表すブロック図。

【図3】本発明を具体化した第3実施形態の概略構成を表すブロック図。

【図4】従来の形態の概略構成を表すブロック図。

【図5】従来の形態の概略構成を表すブロック図。

【図6】従来の形態の動作を表すタイミングチャート。

【図7】従来の形態の概略構成を表すブロック図。

【符号の説明】

2…行列要素設定部 3…演算器群  $P0 \sim Pn$ ,

14…演算器

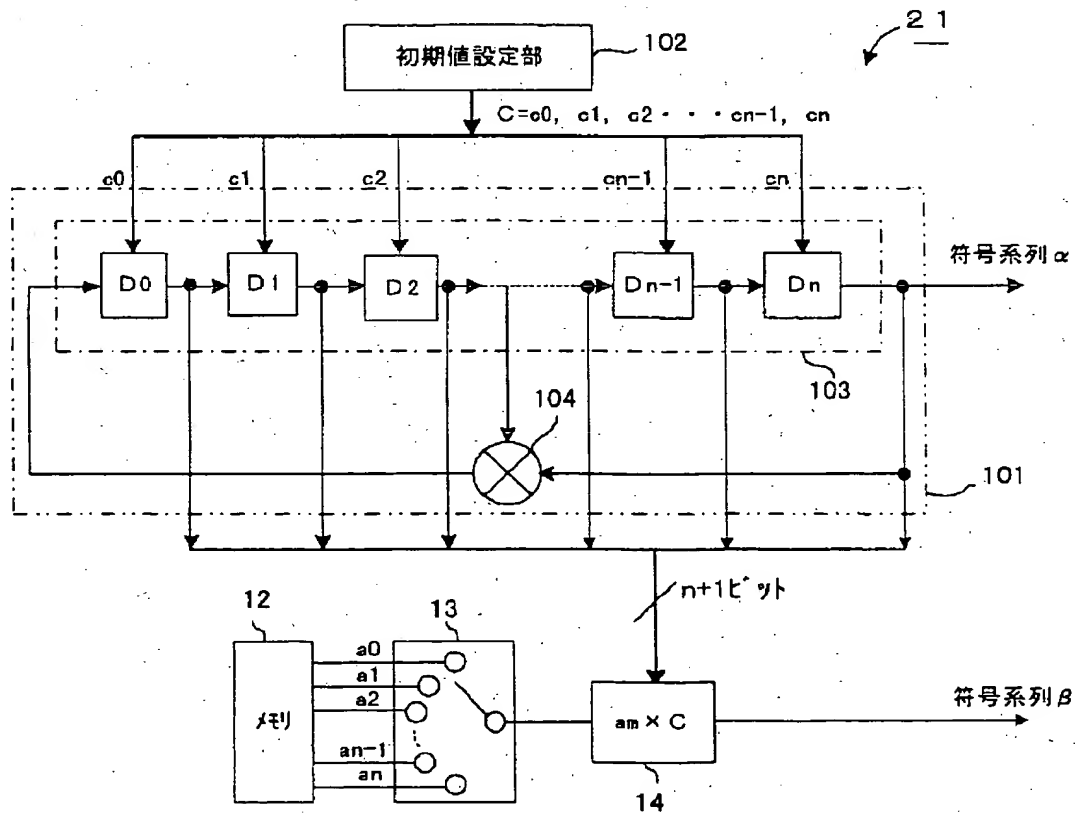
12…メモリ 13…読出回路 15…切替回路

101…符号系列生成器 101a…第2符号系列生成器

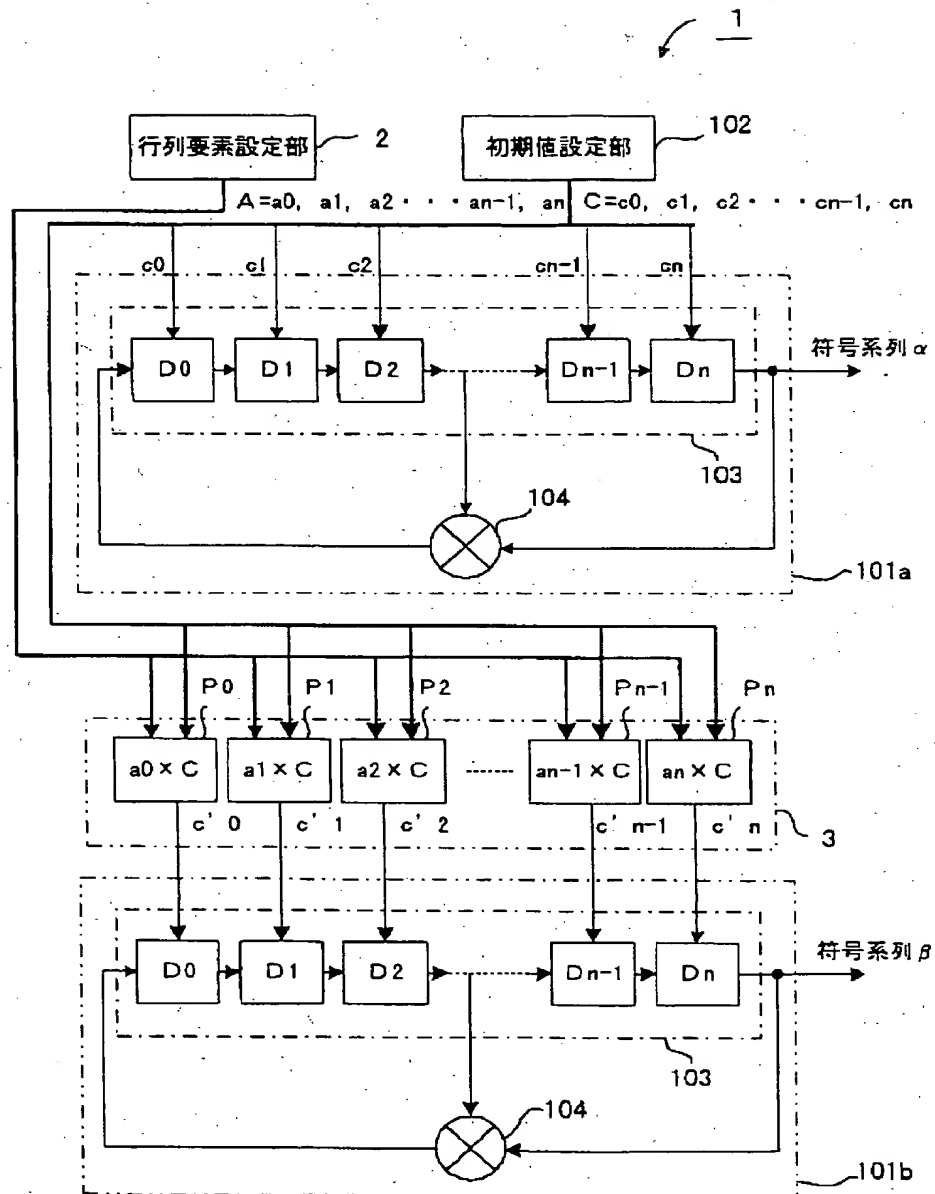
101b…第1符号系列生成器 103…シフトレジスタ

104…排他的論理和回路

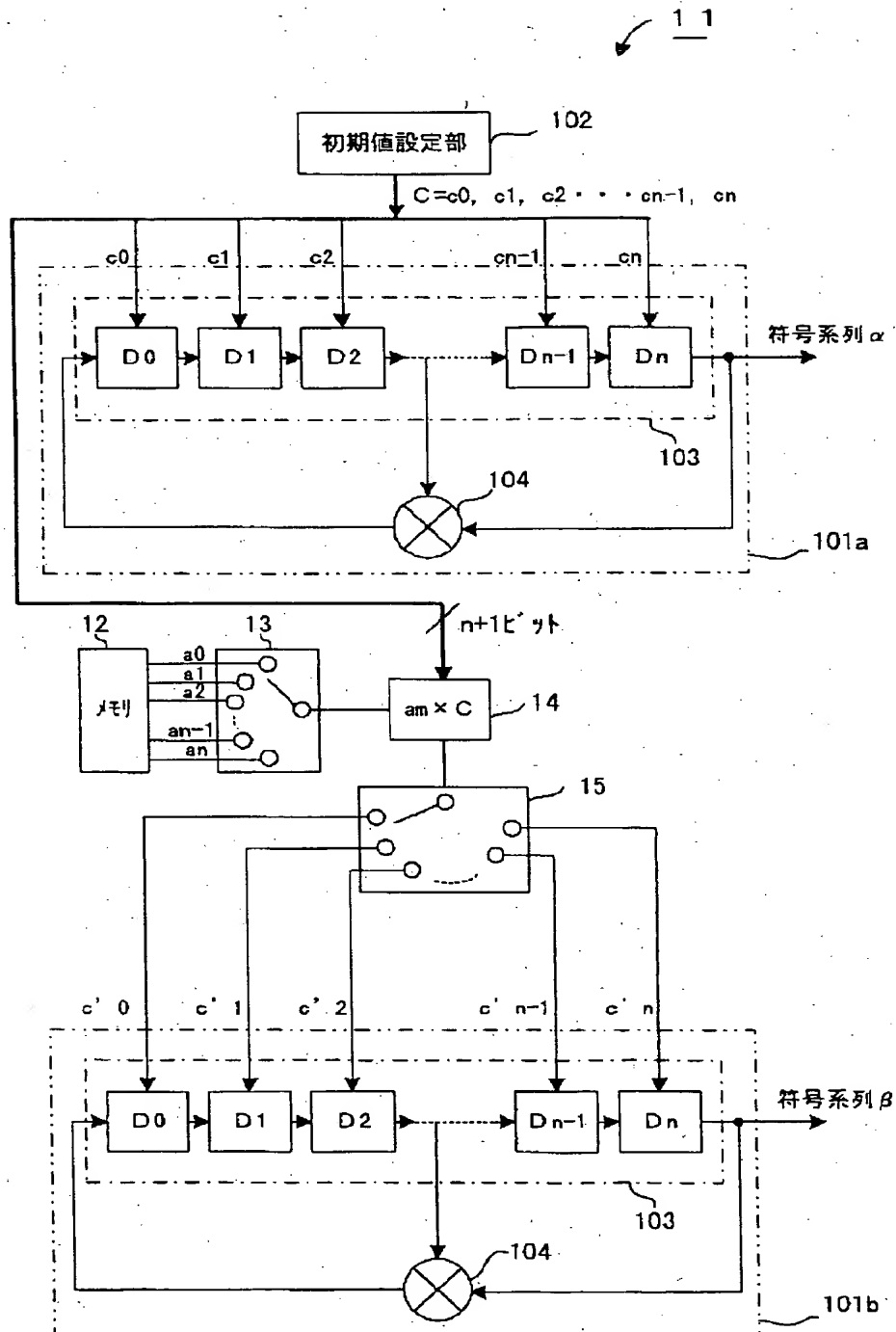
【図3】



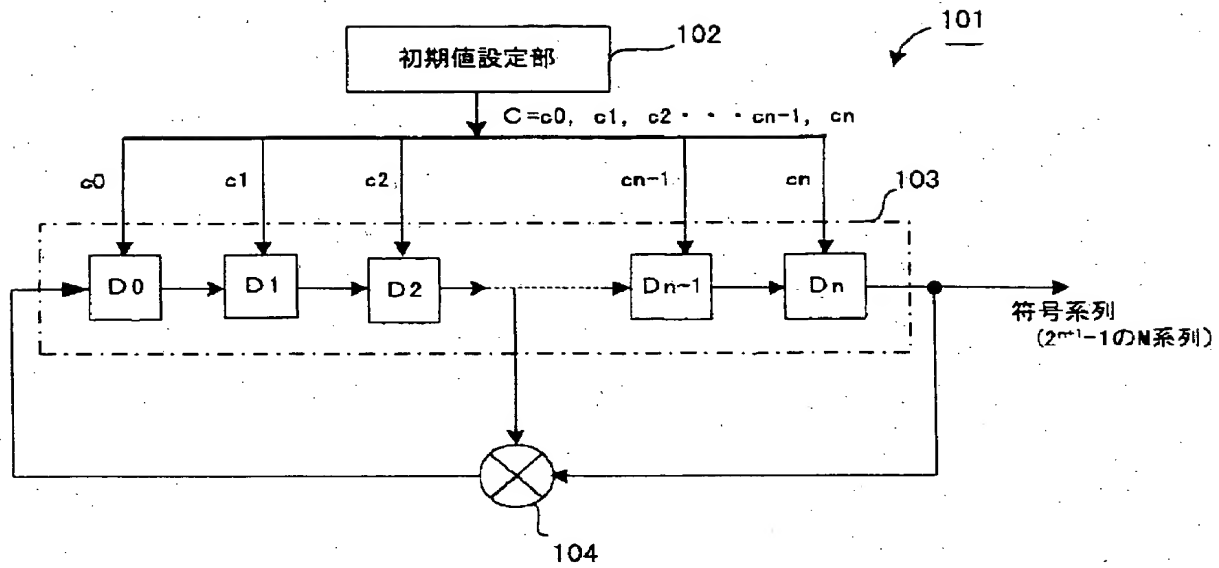
【図1】



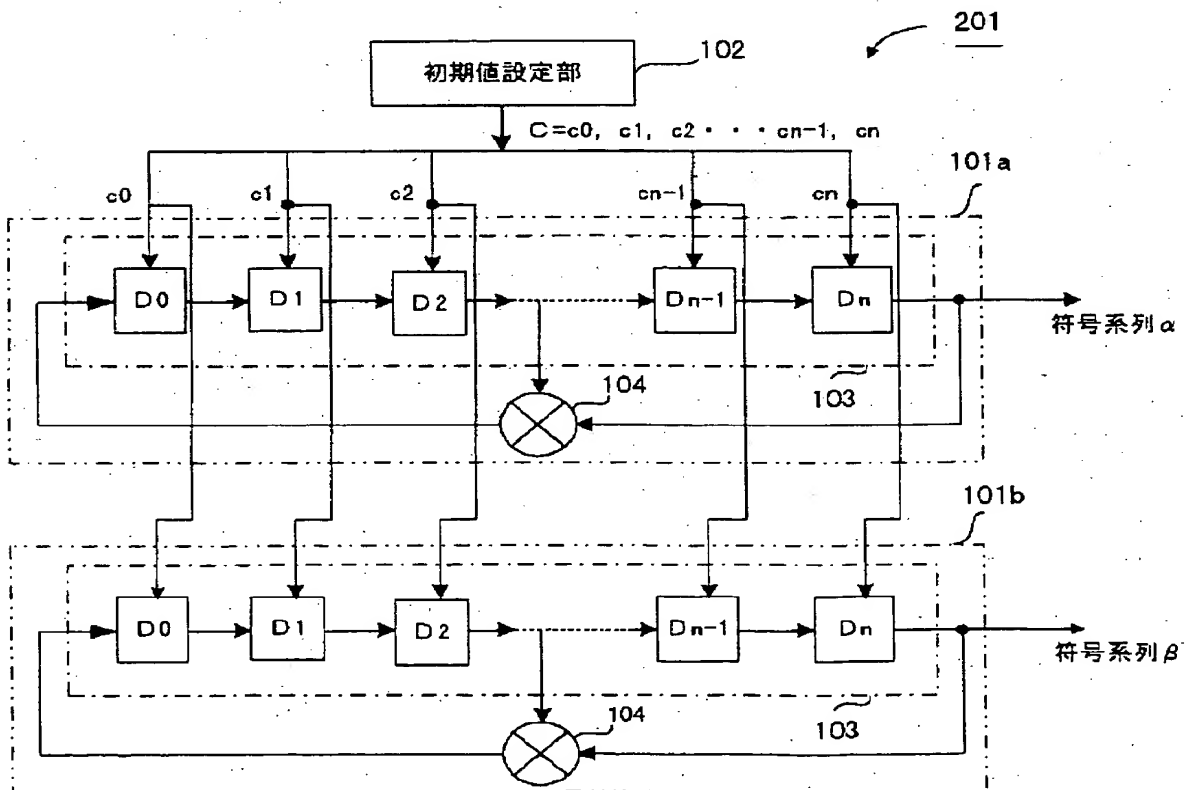
【図2】



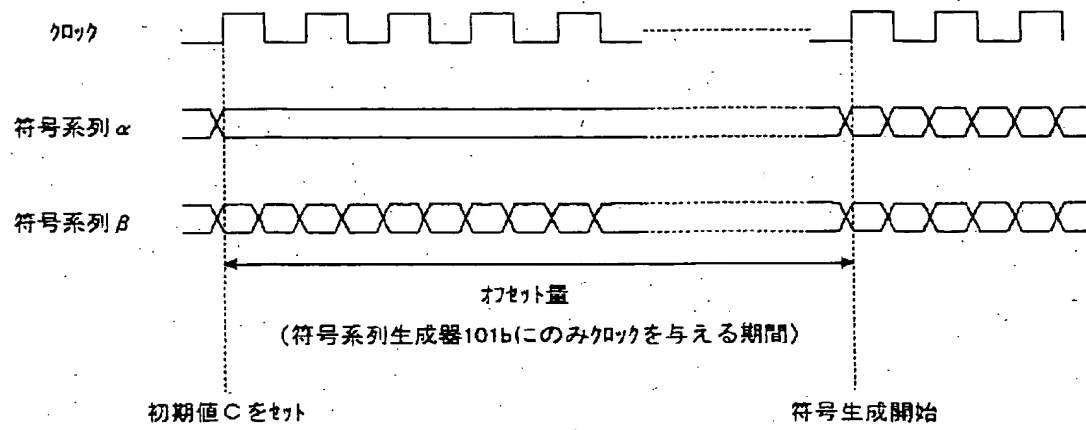
【図4】



【図5】



【図6】



【図7】

